# GET INTO PROGRAMMING WITH JavaScript

Axle Barr

## Programming is being in control

If you can answer the following questions, then you can write a computer program:

1. What comes next 3,6,9…?
2. Can you operate a microwave?
3. Can you follow the steps to bake a cake, change a tire or assemble a piece of furniture?

# NOUNS AND ADJECTIVES

**What is data:**

- The way we describe nouns is what gives rise to data
- Data are the adjectives that describe nouns
- An Employee is a noun as is a Product, a Patient a Trip a Task
- We describe an Employee by her name, phone number, salary etc
- In computer programming we use **values** to represent adjectives
  salary = 50K
  first name = Axle
  status = pending

# ORIGINS OF DATA

Application Form

YOUR LOGO HERE

Company Na

## Employment Application

### Applicant Information

Full Name: _____ Date: _____

*Last* *First* *M.I.*

Address: _____

*Street Address* *Apartment/Unit #*

_____

*City* *State* *ZIP Code*

Phone: _____ Email _____

Date Available: _____ Social Security No.: _____ Desired Salary: $ _____

Position Applied for: _____

Are you a citizen of the United States? YES ☐ NO ☐ If no, are you authorized to work in the U.S.? YES ☐

Have you ever worked for this company? YES ☐ NO ☐ If yes, when? _____

Have you ever been convicted of a felony? YES ☐ NO ☐

If yes, explain: _____

### Education

High School: _____ Address: _____

YOUR LOGO HERE

Company Name

**Employment Application**

**Applicant Information**

Full Name: _____ _____ _____ Date: _____

*Last* *First* *M.I.*

Address: _____ _____

*Street Address* *Apartment/Unit #*

_____ _____ _____

*City* *State* *ZIP Code*

Phone: _____ Email _____

Date Available: _____ Social Security No.: _____ Desired Salary: $ _____

Position Applied for: _____

Are you a citizen of the United States? YES ☐ NO ☐ If no, are you authorized to work in the U.S.? YES ☐ NO ☐

Have you ever worked for this company? YES ☐ NO ☐ If yes, when? _____

Have you ever been convicted of a felony? YES ☐ NO ☐

If yes, explain: _____

**Education**

High School: _____ Address: _____

First Name

Last Name

Date

Street

Apartment

City

State

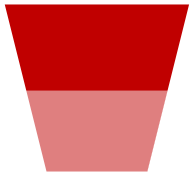
Postal Code

Phone

# Why Use Variables

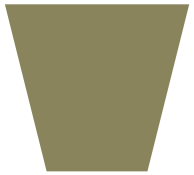
**What is a variable:**

- A *name* that represents a value, just like my *weight* has a value, it is **85** Kgs, my *job* has a value, it is **Trainer**.
- Variables are at the core of every programming language
- A variable is like a bucket, it stores something for use later
- Think of three buckets: red, blue and green



red_bucket

blue_bucket

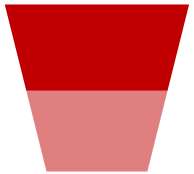


green_bucket

# Named Buckets

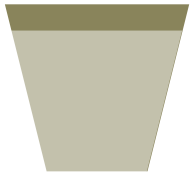
**What if each bucket had a name:**

Let’s say that the red bucket has 300ml of water, the blue has 200 and the green is completely empty



red_bucket

blue_bucket



green_bucket

# Filling Buckets

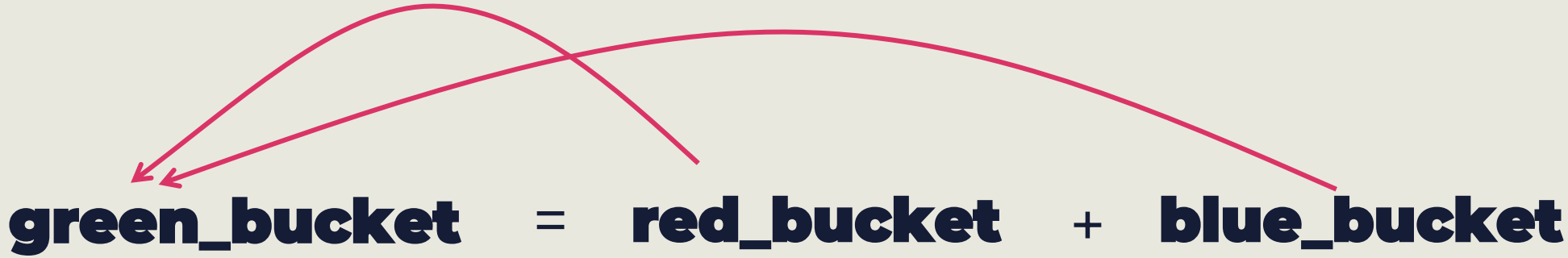
**Pour the contents of the red and blue buckets into the green:**

If the red bucket had 300 ml of water, the blue bucket had 200 ml of water and we poured both buckets of water into the green bucket, we would expect the green bucket to have 500 ml of water.

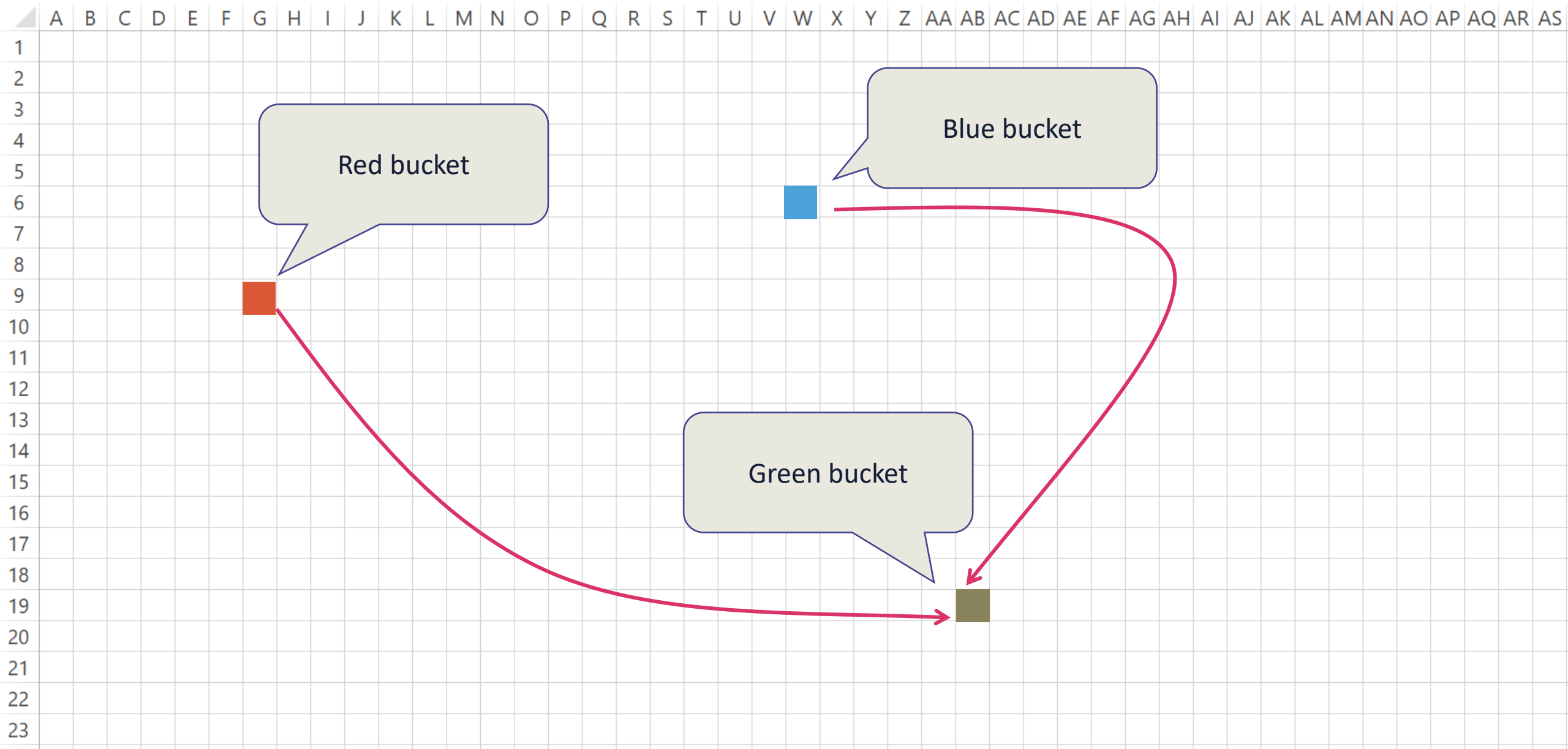
# Algebra: Typical Statement

**red_bucket + blue_bucket = green_bucket**

# Algorithm: Computer Program



# R.A.M



# Algorithm to Program

Translating this to an actual computer program, it will look like this

```
red_bucket = 300
blue_bucket = 200
green_bucket = 0
```

Then, to add the blue and red buckets together we do this:

```
green_bucket = blue_bucket + red_bucket
```

Now, we can printOut green_bucket with a printOut statement

```
printOut green_bucket
```

On the computer screen will be the number **500**

**Computer programs are not like algebra**

# Forget about algebra, for now:

- In algebra, we find statements like these:
  **5 + x = 9**
- And we are supposed to solve for x
- In computers, when we use **x** in any part of our program, **x** represents something - it could be 10, zero or nothing at
- Also remember the **=** sign means *assignment*, so the above statement won't work in programming - it produces an error

# Variable Reuse

What if we did not have a green_bucket?

blue_bucket = red_bucket + blue_bucket

# Variable Reuse

What if we did not have a green_bucket?

red_bucket = red_bucket + blue_bucket

# Algebra vs Computer Program

| | Algebra | Computer Program |
|---|---|---|
| Statement | | |
| Interpretation | | |
| Result | | |

# A Special Variable called Constant:

- Although a variable, as the name suggests, varies, there is one type of variable that does not vary, the **constant** variable
- There are certain values that do not change, such as the number of days in a week, degrees in a circle and wheels on a tricycle.
- Some mathematical constants, like **pi** and **Euler's number** (e), do not change
- We refer to these values in computer science as constant variables (an oxymoron)
- Since these values are needed, we assign the various values to constant variables. For example **pi = 3.142,** normally written as **PI = 3.142**

**The constant variable?**

# Life of a Variable

| Line Number | Statement/Expression | Notes |
|---|---|---|
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |
| | | |

# Specialized Storage

**Use storage based on purpose:**

- A wallet is used for storing paper money
- Only gas goes into the gas tank of your car
- A ceramic cup for hot drinks, but a glass for cold drinks

**Each variable is stored in a type of bucket**

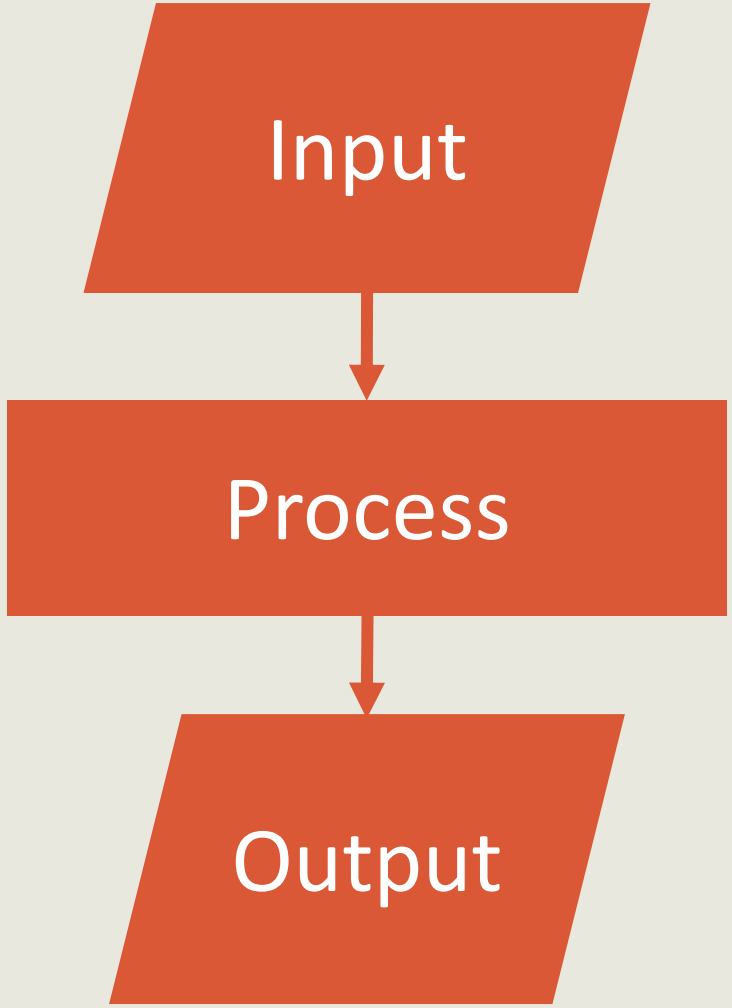
# Variable Types (primitive):

- Computers use memory (RAM) space to store the value that a variable represents
- Some variables require a small amount of bytes others require a larger amount of bytes
- Names have been given for these various types – **integer(number)** for example is used for whole numbers like 0, 200, 50000, -87
- One or more characters are called **string** variables or simply **strings** e.g. “Axle”
- There is a special type of variable called the **Boolean -** it can represent a **yes** or **no**, a **true** or **false** or an **on** or **off**
- Boolean, Null, Undefined, Number, BigInt, String, Symbol

# Typical Variable Names and Types

| Business | Variable Name | Type |
| --- | --- | --- |
| • Hospital | • patientName | • String |
| • Vehicle | • gasTankLitres | • Integer (number) |
| • Government | • criminal_record | • Boolean |
| • Bank | • accountBalance | • Float (number) |

# Input -> Process -> Output



```
let x = 7
```
Take a value of **7** and store that value in a memory location known as **x**

```
let y = 3
```
Take a value of **3** and store that value in a memory location known as **y**

```
let z = x + y
```
Perform the expression of **x+y**, then store the sum in a location called **z**
Same as saying execute **x+y** and assign the resulting value of **10** in **z**
Also same as evaluate the expression on the right side of the assignment operator, then store result in the variable on the left side of the assignment operator

```
printOut(z)
```
Use the printOut() function to find the value stored in a memory location known as **z** and pass it on to the monitor

```
//comment
```
This last line here is a comment, it starts with //

# Fahrenheit to Celsius

$$(F - 32) \times 5/9 = C$$

```
let fahrenheit = 32;
let celsius = 0;
const OFFSET = 32;
celsius = (fahrenheit - OFFSET) * (5/9);
let printout = "The answer is “ + celsius;
```

# Shopping Cart-Final Price

Cost 100
Tax 8% (1.08)

Final price is:
100 * 0.08 = $108.00

```
let productCost = 100;
const TAX = 1.08;
let finalPrice = productCost * TAX;
printout = finalPrice;
```

# INPUT -> PROCESS -> OUTPUT

Input

Process

Output